WHAT IS CLAIMED IS:

1. A method for selectively deploying enterprise software comprising:

for each deployable software component in an preselected input archive file, comparing interfaces for the deployable software component identified in a first descriptor file in said input archive file and a second descriptor file in a preselected output archive file;

if the comparing step miscompares for a first deployable software component, tagging said first deployable software component;

if the comparing step miscompares for a second deployable software component, tagging said second deployable software component; and

deploying each tagged deployable software component.

2. The method of claim 1 wherein tagging a deployable software component comprises storing a name of the deployable software component in a file.

3. The method of claim 1 further comprising:

if the first descriptor file and second descriptor file compare for the first deployable software component, comparing a size of a binary class file for the first deployable software component in the input and output archive files; and

if the size of said binary class files miscompare, tagging the first deployable software component.

4. The method of claim 1 further comprising:

if the first descriptor file and second descriptor file compare for the first deployable software component, introspecting a binary class file for the first deployable software component in the input and output archive files; and

if, in response to the introspection, a signature or return type of an interface of said binary class files miscompare, tagging the first deployable software component.

5. The method of claim 1 further comprising:

opening said preselected output archive file; and

if the step of opening the preselected output archive fails, tagging each deployable software component in the input archive file.

6. The method of claim 5 wherein the step of tagging each deployable software component is performed in response to the step of opening the preselected output archive throwing an exception.

7. The method of claim 1 wherein the comparing, tagging and deploying steps are performed in response to an execution of a build script invoking a selective deployer utility.

8. A computer program product embodied in a machine-readable medium for selectively deploying enterprise software, the program product comprising programming instructions for:

for each deployable software component in an preselected input archive file, comparing interfaces for the deployable software component identified in a first descriptor file in said input archive file and a second descriptor file in a preselected output archive file;

if the comparing operation miscompares for a first deployable software component, tagging said first deployable software component;

if the comparing operation miscompares for a second deployable software component, tagging said second deployable software component; and

deploying each tagged deployable software component.

9. The program product of claim 8 wherein tagging a deployable software component comprises storing a name of the deployable software component in a file.

10. The program product of claim 8 further comprising programming instructions for:

if the first descriptor file and second descriptor file compare for the first deployable software component, comparing a size of a binary class file for the first deployable software component in the input and output archive files; and

if the size of said binary class files miscompare, tagging the first deployable software component.

11. The program product of claim 8 further comprising programming instructions for:

if the first descriptor file and second descriptor file compare for the first deployable software component, introspecting a binary class file for the first deployable software component in the input and output archive files; and

if, in response to the introspection, a signature or return type of an interface of said binary class files miscompare, tagging the first deployable software component.

12. The program product of claim 8 further comprising programming instructions for:

opening said preselected output archive file; and

if the operation of opening the preselected output archive fails, tagging each deployable software component in the input archive file.

13. The program product of claim 12 wherein the operation of tagging each deployable software component is performed in response to the operation of opening the preselected output archive throwing an exception.

14. The program product of claim 8 wherein the comparing, tagging and deploying operations are performed in response to an execution of a build script invoking a selective deployer utility.

15. A data processing system for selectively deploying enterprise software comprising:

circuitry operable for, for each deployable software component in an preselected input archive file, comparing interfaces for the deployable software component identified in a first descriptor file in said input archive file and a second descriptor file in a preselected output archive file;

circuitry operable for, if the comparing operation miscompares for a first deployable software component, tagging said first deployable software component;

circuitry operable for, if the comparing operation miscompares for a second deployable software component, tagging said second deployable software component; and

circuitry operable for deploying each tagged deployable software component.

16. The data processing system of claim 15 wherein tagging a deployable software component comprises storing a name of the deployable software component in a file.

17. The data processing system of claim 15 further comprising:

circuitry operable for, if the first descriptor file and second descriptor file compare for the first deployable software component, comparing a size of a binary class file for the first deployable software component in the input and output archive files; and

circuitry operable for, if the size of said binary class files miscompare, tagging the first deployable software component.

18. The data processing system of claim 15 further comprising programming instructions for:

circuitry operable for, if the first descriptor file and second descriptor file compare for the first deployable software component, introspecting a binary class file for the first deployable software component in the input and output archive files; and

circuitry operable for, if, in response to the introspection, a signature or return type of an interface of said binary class files miscompare, tagging the first deployable software component.

19. The data processing system of claim 15 further comprising

circuitry operable for opening said preselected output archive file; and

circuitry operable for, if the operation of opening the preselected output archive fails, tagging each deployable software component in the input archive file.

20. The data processing system of claim 19 wherein the operation of tagging each deployable software component is performed in response to the operation of opening the preselected output archive throwing an exception.